

## **О МЕТОДАХ ОТЛАДКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МНОГОПРОЦЕССОРНОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ**

**<sup>1</sup>Бураншин И.И., <sup>1</sup>Уманский А.Б., <sup>1</sup>Леонтьев А.В.**

<sup>1</sup> *АО «НАУЧНО-ПРОИЗВОДСТВЕННОЕ ОБЪЕДИНЕНИЕ АВТОМАТИКИ им. академика Н.А. Семихатова», Екатеринбург, Россия (620075 Россия, г. Екатеринбург, ул. Мамина-Сибиряка, 145), e-mail: avt@npoa.ru.*

**Аннотация:** В работе определяются проблемы, возникающие при создании и отладке программного обеспечения для многопроцессорных систем. Рассматривается метод автоматического обнаружения ошибок в системах с распределенной памятью, а также метод отладки с использованием интерфейса JTAG. Предлагается методика комплексной отладки ошибок двух родов в многопроцессорных системах управления.

**Ключевые слова:** многопроцессорная система, методы отладки, программное обеспечение

## **SOFTWARE DEBUGGING METHODS OF THE MULTIPROCESSOR COMPUTING SYSTEM**

**<sup>1</sup> Buranshin I.I., <sup>1</sup> Umanskii A.B., <sup>1</sup> Leontiev A.V.**

<sup>1</sup> *"Scientific and Production Association of automatics named after academician N.A.Semikhatov", Yekaterinburg, Russia (620075 Russian Federation, Yekaterinburg, Mamin-Sibiryak St., 145), e-mail: avt@npoa.ru.*

**The paper represents issues in creation and debugging of software for multiprocessor systems. The method of automatic error detection in systems with distributed memory is examined. The debugging method using JTAG is described. The paper also deals with the complex debugging technique of the two types of errors in multiprocessor control systems.**

**Keywords:** multiprocessor system, debugging methods, software

### **Введение**

Ошибки, возникающие в многопроцессорной системе могут быть как систематическими, возникающими в силу неверного проектирования аппаратно-программной части вычислительной системы, так и стохастическими, определяемые жесткими условиями функционирования. Первый род ошибок может быть устранен путем обширного многократного тестирования и отладки всевозможных веток. Второй род ошибок неустраним в силу собственной природы, однако путем детерминирования поведения системы при возникновении нештатной ситуации можно нивелировать влияние внешнего возмущения. Для парирования ошибок обоих родов в многопроцессорной вычислительной системе необходим комплексный аппаратно-инструментальный подход к отладке программного обеспечения.

## **Проблемы в построении программы для многопроцессорной системы**

Архитектура многопроцессорной системы и соответствующее функциональное распределение задач между модулями системы определяет применяемые методики отладки программного обеспечения. Однако для любой системы, включающего в себя обработку параллельных процессов, характерны определенные проблемы, возникающие в ходе создания и отладки [1, 4]:

- недетерминированное чередование обработки процессов (состояния гонок)
- тупики (ожидание процессом события, которое никогда не произойдет)
- сложность прогнозирования отработки функций при недетерминированном чередовании обработки процессов

Соответственно и инструменты для отладки должны учитывать данные особенности.

## **Методы отладки программного обеспечения**

В работе [3] рассматривается метод автоматического обнаружения ошибок в параллельных программах, ориентированных на системы с распределенной памятью, написанных на языке MPI. Обнаружение ошибок производится на основе анализа трассы выполнения программы, представляющую собой последовательность вызовов функций взаимодействия каждого из процессов параллельной программы. Для каждого вызова записывается имя MPI-функции, время и аргументы вызова.

Обнаружение ошибочных состояний производится анализом очереди событий в трассе, при передаче сообщений. Производится проверка истинности тех или иных заранее заданных предикатов. Если обнаруживается нарушение истинности, то выдается соответствующее диагностическое сообщение.

Взаимодействие параллельных процессов в системе описывается графом событий. Вершины графа - события, происходящие в параллельных процессах; ребра - события, отвечающие операциям отправки и соответствующего приема сообщения. Обнаружение взаимной блокировки производится рекурсивной функцией, которая вызывается каждый раз, когда процесс, управляющий мониторингом, получает событие, соответствующее блокирующей операции обмена (возникновение цикла в графе событий).

Достоинства:

- выявление различных типов недетерминизма
- выявление и устранение различных тупиковых ситуаций

Недостатки присущие данному отладчику:

- неконтролируемое увеличение размера трассы при взаимодействии большого количества процессов параллельной программы
- возможно упущение предикатов, которое может привести к ошибочной работе отладочной программы

## **Отладка с использованием интерфейса JTAG**

Интерфейс JTAG применяется для комплексной отладки многопроцессорной системы [5], а также при граничном сканировании при разработке многопроцессорных систем с гомогенной или гетерогенной структурой.

JTAG-эмулятор позволяет пошагово (на уровне машинных команд) выполнять программу, останавливать периферию: в момент перехода в режим останова выполнения программы пользователя JTAG-эмулятор блокирует источник тактовой частоты, управляющий работой центрального процессора и периферийных устройств. Это обеспечивает относительно точное отслеживание модельного времени. JTAG-эмулятор обеспечивает соответствие времени выполнения программы пользователя времени работы периферийных устройств. При останове JTAG-эмулятор предоставляет доступ ко всем ресурсам микроконтроллера в режиме останова выполнения программы пользователя. Поскольку информация вводится в отлаживаемое устройство и выводится из него через последовательный порт, метод JTAG не может служить заменой полнофункциональному эмулятору, так как не в состоянии отразить текущее состояние внутренних шин.

Однако он имеет ряд очевидных достоинств:

- низкая стоимость средства отладки: в простейшем случае JTAG-эмулятор может состоять из кабеля, соединяющего порт компьютера с JTAG-портом на отлаживаемой плате
- максимально точное соответствие условий отладки рабочим условиям серийного изделия. Как следствие, электрические и временные характеристики системы в процессе отладки абсолютно идентичны характеристикам рабочего режима
- возможность одновременного тестирования нескольких устройств, объединенных JTAG-цепочкой, причем стандарт не вводит никаких ограничений на количество устройств в цепочке

JTAG широко используется для отладки реальных устройств. Тестируемая программа может быть той окончательной версией, которая будет поставляться.

Интерфейс JTAG применяется не только для тестирования, но и для программирования различных устройств, в том числе и энергонезависимой памяти микроконтроллеров.

### **Комплексная отладка многопроцессорных систем в «НПО автоматики»**

Цифровые вычислительные системы, разрабатываемые в АО «НПО автоматики», строятся по магистрально-модульному принципу и предназначены для использования в составе систем управления жесткого реального времени. Модули соединены общесистемной магистралью, по которой производится обмен информацией. Каждый модуль включает в себя несколько процессоров с общей, либо распределенной локальной памятью [2]. Исходя из архитектуры вычислительной системы, создание и отладка программного обеспечения сводится к следующим этапам:

- создание и автономная отладка программного обеспечения в виртуальной платформе
- отладка программного обеспечения на макете вычислительной системы с имитаторами внешнего окружения

– отладка программного обеспечения на цифровой модели объекта управления в реальном времени

Однако следует понимать, что отработка программного обеспечения в виртуальной среде не может гарантировать невозникновения нештатных ситуаций в реальном приборе, поэтому отработке на макете прибора должно уделяться гораздо большее внимание.

Для отработки ошибок первого рода применяется синхронный аппаратный останов, осуществляемый с помощью специализированной ПЛИС, анализирующей шину адреса узла системы, либо на основе отладочных портов каждого блока системы (JTAG). В момент останова появляется возможность списывать журнал профилировщика ядра, трассирующего ход выполнения программы, соответственно по нему восстанавливается последовательность системных вызовов. При обнаружении ошибки в поведении, по графу вызовов анализируются причины возникновения подобной ситуации.

Отработка ошибок второго рода производится, в том числе с помощью синхронного останова. Нештатная ситуация имитируется в каналах резервированных систем, путем например искажения ячейки памяти или регистра в одном из узлов вычислительной системы. Последующий запуск системы с точки останова продемонстрирует реакцию системы на якобы внешнее возмущение. Максимально объемное проставление меток на характерных участках исходных кодов и соответственно имитация неисправной ситуации в каждой из этих меток позволит проанализировать поведение системы при реальном возникновении сбоя в рантайме.

## **Заключение**

Применение рассмотренных отладчиков позволяет решать ряд локальных проблем, возникающих в процессе разработки многопроцессорной системы. Внедрение современных программных средств и новых подходов в комплексную отладку позволяет повысить вероятность исключения всевозможных отрицательных веток в программном обеспечении. При этом целью комплексной отладки ставится выявление не только ошибок проектирования архитектуры различного рода, но и определения поведения программы при воздействии внешних факторов на аппаратуру системы управления.

## **Список литературы**

1. Murillo, L.G. Debugging Concurrent MPSoC Software with Bug Pattern Descriptions // System, Software, SoC and Silicon Debug Conference (S4D'11). – 2011.
2. Антимиров, В.М. Бортовые цифровые вычислительные системы семейства «Малахит» для работы в экстремальных условиях / В.М. Антимиров, А.Б. Уманский, Л.Н. Шалимов // Вестник СГАУ. 2013. №4(42). С.19-27.
3. Седов, Д.А. Алгоритмы обнаружения ошибок в параллельных программах для систем с распределенной памятью / Д.А. Седов // Труды ИСА РАН. Том 63. 4/2013. С.9-15.
4. Галатенко, В.А. К проблеме отладки разнородных многопроцессорных комплексов / В.А. Галатенко, К.А. Костюхин, Н.В. Шмырев // [Электронный ресурс]. — Режим доступа: [http://cmm.ipu.ru/sites/default/cmm12cd/CD/Papers/galatenko\\_pdfed\\_.pdf](http://cmm.ipu.ru/sites/default/cmm12cd/CD/Papers/galatenko_pdfed_.pdf) (дата обращения: 01.11.2015).
5. Методы и средства отладки микропроцессорных систем. Национальный Открытый Университет «ИНТУИТ» [Электронный ресурс]. — Режим доступа:

<http://www.intuit.ru/studies/courses/604/460/lecture/10353?page=4> (дата обращения: 25.11.2015).

## References

1. Murillo L. G. Debugging Concurrent MPSoC Software with Bug Pattern Descriptions // System, Software, SoC and Silicon Debug Conference (S4D'11). – 2011.
2. Antimirov V.M. Onboard digital computing systems of Malakhit family for work in extreme conditions / V. M. Antimirov, A.B. Umansky, L.N. Shalimov // The Samara State Aerospace University Bulletin. 2013. No. 4(42). Page 19-27.
3. Sets D. A. Algorithms of detection of mistakes in parallel programs for systems with the distributed memory. Works of Institute for Systems Analysis of the Russian Academy of Sciences. Volume 63. 4/2013. Page 9-15.
4. Galatenko V.A., Kostyukhin K.A., Shmyrev N. V. To a problem of debugging of heterogeneous multiprocessor complexes [An electronic resource]. URL: [http://cmm.ipu.ru/sites/default/cmm12cd/CD/Papers/galatenko\\_pdfed\\_.pdf](http://cmm.ipu.ru/sites/default/cmm12cd/CD/Papers/galatenko_pdfed_.pdf) (date of the address: 01.11.2015).
5. Methods and tools for debugging microprocessor systems. National Open University "INTUIT" [An electronic resource]: URL: <http://www.intuit.ru/studies/courses/604/460/lecture/10353?page=4> (date of the address: 25.11.2015).